MATHEMATICS OF COMPUTATION
VOLUME 47, NUMBER 175
JULY 1986, PAGES 323-346

Symmetric FFTs

By Paul N. Swarztrauber*

Abstract. In this paper, we examine the FFT of sequences $x_n = x_{N+n}$ with period N that satisfy certain symmetric relations. It is known that one can take advantage of these relations in order to reduce the computing time required by the FFT. For example, the time required for the FFT of a real sequence is about half that required for the FFT of a complex sequence. Also, the time required for the FFT of a real even sequence $x_n = x_{N-n}$ requires about half that required for a real sequence. We first define a class of five symmetries that are shown to be "closed" in the sense that if x_n has any one of the symmetries, then no additional symmetries are generated in the course of the FFT. Symmetric FFTs are developed that take advantage of these intermediate symmetries. They do not require the traditional pre- and postprocessing associated with symmetric FFTs and, as a consequence, they are somewhat more efficient and general than existing symmetric FFTs.

1. Introduction. Let $x_n = x_{n+N}$ be a complex periodic sequence with period N. Then x_n has the characterization

$$x_n = \sum_{k=0}^{N-1} X_k e^{ikn2\pi/N}, \tag{1.1}$$

where

$$X_k = \frac{1}{N} \sum_{n=0}^{N-1} x_n e^{-ikn2\pi/N}. \tag{1.2}$$

The sequence X_k is the discrete Fourier transform of x_n, and x_n is the inverse discrete Fourier transform of X_k.

It is well known that the transforms (1.1) and (1.2) can be computed more efficiently using the fast Fourier transform (FFT). It is also known that further economies in computation are possible for sequences that satisfy one or more relations in addition to the periodic relation $x_n = x_{n+N}$. For example, if $x_n = \bar{x}_n$ is a real sequence, then X_k can be computed in half the time that is required for a complex sequence. If the real sequence is also even, then $x_n = x_{N-n}$ and the amount of computation can again be halved. Sequences that satisfy relations of this type are said to be symmetric and variants of the FFT that take advantage of these symmetries are called symmetric FFTs.

The symmetric FFTs are used extensively in the solution of boundary-value problems in partial differential equations. Originally, the Fourier method could only be used for problems in which periodic boundary conditions were specified [6]. Later

Received April 11, 1984; revised February 8, 1985 and July 31, 1985.
1980 *Mathematics Subject Classification*. Primary 42-04, 65F05; Secondary 65T05, 65P05.
Key words and phrases. FFT, fast Fourier transform, symmetric FFTs.
*This work was sponsored by the National Science Foundation.

the sine and cosine symmetric transforms were used for Dirichlet-Dirichlet and Neumann-Neumann boundary conditions, respectively. More recently, the Fourier method has been applied to Dirichlet-Neumann and Neumann-Dirichlet boundary-value problems using the symmetric quarter-wave transforms [7], [11]. The Fourier method can also be used for certain problems that are posed on a staggered grid. The algorithms presented here further extend the Fourier method to problems in which one boundary is at a grid point and the other is halfway between two grid points. This is typically the case for Dirichlet-Neumann boundary conditions.

The fast symmetric Fourier transforms have also been used for computing the associated Legendre functions [9]. The coefficients in the trigonometric representations of the functions are computed efficiently as the solution of a tridiagonal system of equations. The function can then be tabulated using a symmetric FFT. The result is a reliable and accurate method for computing the associated Legendre functions which, unlike methods that use recurrence relations, allows a single function of any degree and order to be computed without computing functions of adjacent degree or order.

Most of the existing algorithms for the symmetric FFTs are based on preprocessing the symmetric sequence into a complex or real sequence which can then be transformed using programs for the complex or real transform that are readily available. The results of this transform are then postprocessed, or decoded, in order to obtain the symmetric transform. This approach is satisfactory for most applications, but it does have certain nuisance attributes. First, the length N of the sequence must be an even integer, or there must be an even number of sequences. Second, because the FFT is itself a very efficient algorithm, the pre- and postprocessing of the sequence take a noticeable amount of computing time. The purpose of this paper is to present new algorithms that are somewhat more efficient and general than their predecessors. The traditional pre- and postprocessing associated with symmetric FFTs are eliminated, together with the restriction that N be an even integer.

The algorithms presented here are obtained by modifying the FFT itself. They are derived by identifying the intermediate symmetries that occur in the FFT of a symmetric sequence and using these symmetries to eliminate duplicate or zero computations. Hence, the computations are the same as those in the full complex transform of a symmetric sequence, but fewer in number, with the result that the stability properties are at least as good as those of the complex transform.

We find that an even sequence induces intermediate sequences within the FFT that have real periodic, real even, and real even quarter-wave symmetries. An odd sequence induces real periodic, real odd, and real odd quarter-wave symmetries. Further, we find that these five symmetries are "closed" in the sense that if the original sequence has any one of the symmetries, then each of the intermediate sequences within the FFT also has one of the five symmetries. This "closure" property is the reason we restrict our attention to only these symmetries. The real even and odd sequences are represented as a cosine and sine series, respectively. The even and odd quarter-wave transforms are represented as cosine and sine series, respectively, but with odd wave numbers only.

There are three fundamental steps in the development of the symmetric FFTs that are given in this paper. First, the intermediate symmetries generated in the course of the FFT must be identified. Second, the intermediate symmetries in the transforms which are induced by the symmetries in the sequences, must be identified. And third, these symmetries must be used to eliminate unnecessary computations in the FFT.

In Section 2, we identify the symmetries both in the intermediate sequences and in their transforms for the case where N is a power of 2. In Section 3, we use these symmetries to develop the symmetric FFTs. In this paper, the symmetric FFTs are developed in the context of the Cooley-Tukey FFT. However, they could also be developed in the context of any FFT algorithm including the Pease or Stockham autosort algorithms. In Section 4, the results of Sections 2 and 3 are extended to the case where N is a product of primes. In Section 5, we present the existing symmetric FFTs for comparison with those developed in this paper. Finally, in Section 6, we compare operation counts with the existing symmetric FFTs and discuss certain aspects of implementation, including the vectorization of the algorithms.

2. The Symmetries.

2.1. *Preliminaries.* Since the complex transform (1.2) can be used to compute the transform of any periodic complex sequence, it is evident that it could be used to compute the transform of a symmetric sequence. However, this would be inefficient since it is known that symmetric sequences can be transformed with fewer operations than required by (1.2). Existing symmetric FFTs have been collected and presented in Section 5. The algorithms presented here are derived by modifying the FFT. In order to develop notation and provide a basis upon which to describe the symmetric FFTs, it is necessary to review the FFT. In particular, we will present a detailed derivation of the Cooley-Tukey FFT.

2.2. *The Cooley-Tukey FFT.* The Cooley-Tukey FFT is a straightforward extension of the following simple splitting algorithm. The motivation for the splitting algorithm is simply that it halves the computation in comparison with that required by either (1.1) or (1.2). If N is even, the sum on the right side of (1.2) can be divided into two sums in which the subscripts are even and odd:

$$X_k = \sum_{n=0}^{N/2-1} x_{2n} e^{-ik2n2\pi/N} + \sum_{n=0}^{N/2-1} x_{2n+1} e^{-ik(2n+1)2\pi/N}. \tag{2.1}$$

It is customary in the development of the FFT to ignore the scale factor of $1/N$ in (1.2). If we define

$$Y_k = \sum_{n=0}^{N/2-1} x_{2n} e^{-ikn2\pi/(N/2)}, \tag{2.2}$$

$$Z_k = \sum_{n=0}^{N/2-1} x_{2n+1} e^{-ikn2\pi/(N/2)}, \tag{2.3}$$

then (2.1) has the form

$$X_k = Y_k + e^{-ik2\pi/N} Z_k, \qquad k = 0, \dots, N/2-1. \tag{2.4}$$

From (2.2) and (2.3), it can be determined that Y_k and Z_k have the period $N/2$. That is, $Y_{k+N/2} = Y_k$ and $Z_{k+N/2} = Z_k$. Also, since

$$e^{-i(k+N/2)2\pi/N} = -e^{-ik2\pi/N}, \tag{2.5}$$

we have

$$X_{k+N/2} = Y_k - e^{-ik2\pi/N} Z_k, \qquad k = 0, \dots, N/2-1. \tag{2.6}$$

The splitting algorithm for computing X_k can now be given.

1. Compute Y_k and Z_k from Eqs. (2.2) and (2.3), which requires $N^2/2$ complex multiplications and $N(N/2 - 1)$ complex additions.

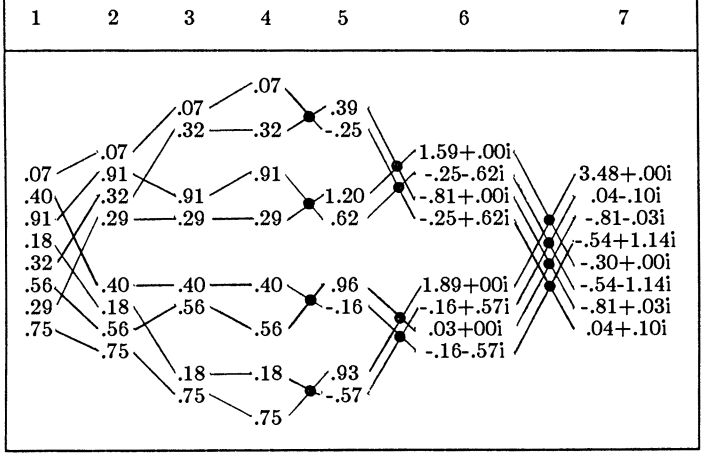
2. Compute X_k from (2.4) and (2.6), which requires $N/2$ multiplications, $N/2$ additions, and $N/2$ subtractions.

The motivation for the splitting algorithm is now evident. For large N, the computation in step 2 is negligible compared to that in step 1. Therefore, the total number of operations required to compute X_k using the splitting algorithm is about half the number required to compute X_k directly using (1.2).

The Cooley-Tukey FFT algorithm consists of repeated application of the splitting algorithm in the following manner. We note that Y_k and Z_k, as defined in (2.2) and (2.3), have the same form as X_k, as defined in (1.2), except that N has been replaced with $N/2$. Therefore, the computing time can be further reduced by using the splitting algorithm to compute both Y_k and Z_k. If $N = 2^m$, then the splitting algorithm can be applied m times with the result that the FFT requires $mN/2 = (N/2)\log_2 N$ complex additions, subtractions, and multiplications.

The Cooley-Tukey algorithm for a random real sequence of length $N = 2^3 = 8$ is given in Table 1. The original sequence x_n is in column 1 and its transform is given in column 7. In column 2, the sequence has been split into two sequences, x_{2n}, x_{2n+1}, with even and odd subscripts, respectively. Further splittings are given in columns 3 and 4. Since, by definition (1.2), the transform of a sequence of length 1 is just itself, column 4 also contains the transforms. Four transforms of length 2 are computed in column 5. Each transform is computed from two transforms of length 1 in column 4, using Eqs. (2.4) and (2.6) with $N = 2$. Next, two transforms of length 4 are computed in column 6. Each transform is computed from two transforms of length 2 in column 5, using (2.4) and (2.6) with $N = 4$. These transforms are Y_k and Z_k, respectively, which were defined in (2.2) and (2.3). Finally, X_k is computed in column 7 from Y_k and Z_k in column 6, using (2.4) and (2.6), with $N = 8$.

TABLE 1
The Cooley-Tukey FFT for N = 8



1	2	3	4	5	6	7
.07	.07	.07	.07	.39	1.59+.00i	3.48+.00i
.40	.91	.32	.32	-.25	-.25-.62i	.04-.10i
.91	.32	.91	.91	1.20	-.81+.00i	-.81-.03i
.18	.29	.29	.29	.62	-.25+.62i	-.54+1.14i
.32	.40	.40	.40	.96	1.89+00i	-.30+.00i
.56	.18	.56	.56	-.16	-.16+.57i	-.54-1.14i
.29	.56	.18	.18	.93	.03+00i	-.81+.03i
.75	.75	.75	.75	-.57	-.16-.57i	.04+.10i

Note that each of the sequences in columns 5, 6, and 7 are the FFTs of the corresponding sequences in columns 3, 2, and 1, respectively. This observation leads to a symmetric transform for real sequences. From (1.2) it is straightforward to show that the transform of a real sequence is conjugate symmetric, i.e., $X_k = \overline{X}_{N-k}$. But this implies that all of the transforms in columns 5, 6, and 7 are conjugate symmetric because they are the transforms of the real sequences in columns 3, 2, and 1. Therefore, only half of the transforms in columns 5, 6, and 7 need to be computed. The resulting algorithm was published by Bergland [1], who credits Edson as its originator. It requires a little less than half the operations of the complex FFT. Temperton reports [12] that it is also approximately 30% faster than the existing symmetric transforms for a real sequence which use pre- and postprocessing. It is also free of the nuisance restrictions that N be even or that the number of transforms be even.

The concept of exploiting the symmetries in the intermediate sequences that are generated in the FFTs has motivated the approach that is taken in this paper. The algorithms for the remaining symmetric sequences are developed by identifying the symmetries that are induced in the intermediate sequences and transforms. Unlike the development for real sequences, the development for the remaining symmetries is complicated by the fact that new symmetries are generated in the course of the FFT. In the remainder of this section, we will determine these intermediate symmetries.

2.3. *Symmetries Induced by a Symmetric Sequence.* In what follows, it will be convenient to associate a mnemonic with each symmetry. Thus, we define a real even sequence $x_n = x_{N-n}$ as E symmetric. Following the splitting algorithm for real sequences, we split x_n into the sequences $y_n = x_{2n}$ and $z_n = x_{2n+1}$ and ask the question: What symmetries do y_n and z_n have? The answer is obtained by noting that

$$y_{N/2-n} = x_{N-2n} = x_{2n} = y_n, \tag{2.7}$$

and hence y_n is also E symmetric. In addition,

$$z_{N/2-n-1} = x_{N-2n-1} = x_{2n+1} = z_n. \tag{2.8}$$

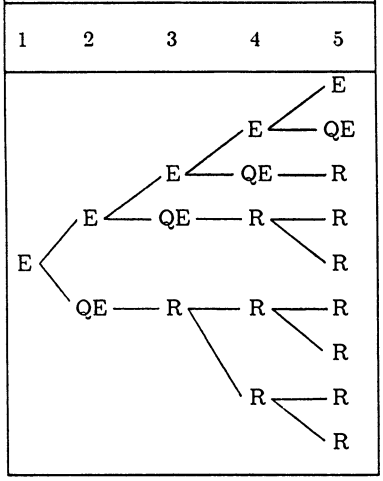
We will define such a sequence as QE symmetric; that is, any sequence x_n such that $x_n = x_{N-n-1}$ is called QE symmetric. The mnemonic QE stands for quarter-wave even. The quarter-wave terminology follows that used in Section 5, and in [7], where quarter-wave transforms are discussed that are QE symmetric as defined here.

The sequence z_n will also be split in the FFT and therefore it is necessary to identify the symmetries that are produced when a QE symmetric sequence is split. Let x_n be QE symmetric, and again define $y_n = x_{2n}$ and $z_n = x_{2n+1}$. Considered individually, these sequences have only real symmetry. However, they do have an intersequence symmetry, namely

$$y_{N/2-n-1} = x_{N-2n-2} = x_{2n+1} = z_n. \tag{2.9}$$

For this reason, we can ignore one of these sequences, say z_n, and observe simply that a QE symmetric sequence of length N splits into a single R (real) symmetric sequence y_n of length $N/2$. We see now that the symmetries R, E, and QE are closed in the sense that a sequence with any one of these symmetries splits into two sequences, each of which also has one of the three symmetries.

TABLE 2
A list of symmetries induced by an even sequence of length $N = 16$



The splittings and symmetries for an E symmetric sequence are given in Table 2 for the case $N = 2^4 = 16$. The original sequence of length 16 is designated by a single E in the first column. At the first step of the FFT the original sequence is divided into two sequences, each of length 8. From the results above, the first sequence is E symmetric and the second is QE symmetric. These sequences are designated by the E and QE that appear in the second column of Table 2. The subsequent splittings are shown in the remaining columns of the table. Note that a QE symmetric sequence of length $N/2^i$ splits into a single R symmetric sequence of length $N/2^{i+1}$ as described above.

If we define a real odd sequence $x_n = -x_{N-n}$ as O symmetric and proceed in a manner similar to that above, we can show that an O symmetric sequence splits into an O symmetric sequence and a QO symmetric sequence $z_n = -z_{N/2-n-1}$.

This completes the identification of the symmetries in the intermediate sequences.

2.4. *Symmetries in the Transform of the Symmetric Sequences.* Recall that the development of Edson's algorithm for an R symmetric sequence depended on the fact that the transform of a real sequence was conjugate symmetric. That is, the real FFT depended on the symmetry in the transform as well as the symmetry in the original sequence. Therefore, the next step is to identify the symmetry in the transform that is induced by the symmetry in the sequence.

We will begin this part with a derivation of the symmetry in the transform of a QE symmetric sequence. This derivation is sufficiently representative that the remaining symmetries are presented in Table 3 without derivation.

Let $x_n = x_{N-n-1}$ be a real sequence with QE symmetry. In order to determine the symmetry that is induced in the Fourier transform, we begin by reversing the order of summation on the right side of (1.2), with the result

$$X_k = \sum_{n=0}^{N-1} x_{N-n-1} e^{-ik(N-n-1)2\pi/N}. \tag{2.10}$$

But $x_n = x_{N-n-1}$, and therefore,

(2.11) $$X_k = e^{ik2\pi/N} \sum_{n=0}^{N-1} x_n e^{ikn2\pi/N},$$

or

(2.12) $$X_{N-k} = e^{-ik2\pi/N} \sum_{n=0}^{N-1} x_n e^{-ikn2\pi/N}.$$

From (1.2) and (2.12) we determine that the Fourier transform of a QE symmetric sequence satisfies

(2.13) $$X_k = e^{ik2\pi/N} X_{N-k}.$$

The remaining symmetries are presented in Table 3. Note that the symmetries are first presented for complex sequences. The last four rows are obtained by combining the real symmetry with the complex symmetries. In what follows, we will only discuss the real symmetries since they are by far the most commonly encountered in practice.

TABLE 3
Symmetries

symmetry	sequence	transform
periodic	$x_n = x_{N+n}$	$X_k = X_{N+k}$
complex even	$x_n = x_{N-n}$	$X_k = X_{N-k}$
complex odd	$x_n = -x_{N-n}$	$X_k = -X_{N-k}$
complex qtr. even	$x_n = x_{N-n-1}$	$X_k = e^{ik2\pi/N} X_{N-k}$
complex qtr. odd	$x_n = -x_{N-n-1}$	$X_k = -e^{ik2\pi/N} X_{N-k}$
real (R)	$x_n = \bar{x}_n$	$X_k = \bar{X}_{N-k}$
real even (E)	$x_n = x_{N-n}$	$X_k = \bar{X}_k$
real odd (O)	$x_n = -x_{N-n}$	$X_k = -\bar{X}_k$
qtr. even (QE)	$x_n = x_{N-n-1}$	$X_k = e^{ik2\pi/N} \bar{X}_k$
qtr. odd (QO)	$x_n = -x_{N-n-1}$	$X_k = -e^{ik2\pi/N} \bar{X}_k$

From a computational point of view the quarter-wave symmetries are not as satisfactory as the other symmetries. For example, the transform of an even sequence is strictly real, with the result that complex arithmetic can be eliminated, at least to a certain extent. On the other hand, the transform of a quarter-wave sequence is complex and it is therefore less obvious how one can take advantage of the symmetry. In the remainder of this section, we will show that these transforms can be represented in terms of real (or strictly imaginary) sequences. These representations will then be used in Section 3 in order to reduce the amount of computation in the symmetric transform.

Consider first the symmetry in the transform X_k of a QE symmetric sequence. From Table 3, $X_k = e^{ik2\pi/N}\bar{X}_k$, which implies that the argument of X_k is $\alpha_k = k\pi/N$. Therefore, X_k has the form

(2.14) $$X_k = e^{ik\pi/N}\tilde{X}_k,$$

where $\tilde{X}_k$ is real.

Consider now the symmetry $X_k = -e^{ik2\pi/N}\bar{X}_k$ in the transform of a QO sequence. This symmetry implies that the argument of X_k is $\alpha_k = k\pi/N + \pi/2$, and therefore X_k has the form

$$X_k = e^{ik\pi/N}\tilde{X}_k, \tag{2.15}$$

where $\tilde{X}_k$ is strictly imaginary.

In both Sections 3 and 4, we will make use of these forms in order to replace complex with real arithmetic whenever possible. $\tilde{X}_k$ rather than X_k will always be computed for any quarter-wave transform.

3. The Symmetric FFTs.

3.1. *Preliminaries.* In this section, we will complete the development of the symmetric FFTs by using the symmetries that were identified in the previous section to develop new splitting equations. The FFT itself consists of the repeated application of the splitting equations in combinations that are evident from Table 2 or a similar table which could be presented for any of the symmetric sequences. Only the splitting equations are presented. However, they must be applied repeatedly and in the proper combination in order to define the symmetric FFT. As in the previous section, we will assume that N is an even integer; however, in the next section we will show that the algorithms can be generalized to arbitrary N. We begin this section with a derivation of the splitting equations that are used in Edson's algorithm for the fast transform of a real symmetric sequence.

3.2. *Splitting Equations for R Symmetric Sequences.* The splitting equations for real sequences can be determined from Eqs. (2.4) and (2.6) for complex sequences. Since x_n, y_n, and z_n are real sequences, their transforms are conjugate symmetric, i.e., $X_k = \bar{X}_{N-k}$, $Y_k = \bar{Y}_{N/2-k}$, and $Z_k = \bar{Z}_{N/2-k}$. Therefore, it is only necessary to compute half of these sequences. Assuming that Y_k and Z_k are available for $k = 0, \ldots, N/4$, we can compute the first half of the X_k from (2.4).

$$X_k = Y_k + e^{-ik2\pi/N}Z_k, \qquad k = 0, \ldots, N/4. \tag{3.1}$$

In order to compute X_k for $k = N/4 + 1, \ldots, N/2$ we begin with

$$X_{N/2-k} = Y_{N/2-k} + e^{-i(N/2-k)2\pi/N}Z_{N/2-k}. \tag{3.2}$$

Using the conjugate symmetry of both Y_k and Z_k, we obtain

$$\bar{X}_{N/2-k} = Y_k - e^{-ik2\pi/N}Z_k, \qquad k = 0, \ldots, N/4 - 1. \tag{3.3}$$

Equations (3.1) and (3.3) comprise the splitting algorithm for R symmetric sequences. Hence, the transform of an R symmetric sequence of length N can be computed from the transforms of two R symmetric sequences of length $N/2$ with $N - 4$ real multiplications and $3N/2 - 2$ real additions.

The symmetric FFT for a real sequence which consists of the repeated application of the splitting equations (3.1) and (3.3) is called Edson's algorithm. It requires a little less than half the number of operations of the Cooley-Tukey FFT for a complex sequence (see Table 4). The computations in Edson's real FFT form a subset of those in the Cooley-Tukey algorithm, and therefore, the stability of Edson's algorithm is at least as good as the Cooley-Tukey algorithm. Note also that no pre- or postprocessing is required.

The approach that was taken above will be used to develop efficient algorithms for the remaining symmetric transforms. The key is to identify the symmetries that are present in the intermediate sequences and transforms in columns 2 through 6 in Table 1, when the first column has any of the symmetries that were defined in Section 2.

3.3. *Splitting Equations for E Symmetric Sequences.* Unlike the full complex transform and Edson's FFT for real symmetric sequences, the remaining symmetric FFTs use several different splitting equations. For example, the symmetric FFT of an E symmetric sequence will use equations that correspond to splittings of E, QE, and R symmetric sequences. The order in which the equations are used is evident from Table 2. The splitting equations are all derived from those given above for Edson's real FFT, which was itself derived from the splitting equations for the complex FFT. We continue this approach in which the equations are developed by eliminating duplicate or zero computations.

Consider now an E symmetric sequence $x_n = x_{N-n}$, and its splittings $y_n = x_{2n}$ and $z_n = x_{2n+1}$. As before, we denote their transforms by Y_k and Z_k, respectively. From the results in Section 2, Y_k is also E symmetric and Z_k is QE symmetric. Therefore, both X_k and Y_k are real and, from (2.14), Z_k has the form $Z_k = e^{ik2\pi/N}\tilde{Z}_k$, where $\tilde{Z}_k$ is real. Substituting this form into (3.1) and (3.3), we obtain the splitting equations for an E symmetric sequence,

(3.4) $$X_k = Y_k + \tilde{Z}_k, \qquad k = 0, \ldots, N/4,$$

(3.5) $$X_{N/2-k} = Y_k - \tilde{Z}_k, \qquad k = 0, \ldots, N/4 - 1.$$

Since all the quantities are real, only real arithmetic is needed. Equations (3.4) and (3.5) provide the splitting algorithm for an E symmetric sequence. Note that only $N/2 + 1$ real additions are required. Note also that for any QE symmetric sequence, $\tilde{Z}_k$ is computed rather than Z_k, in order to avoid complex arithmetic.

3.4. *Splitting Equations for O Symmetric Sequences.* Consider now an O symmetric sequence $x_n = -x_{N-n}$ and its splittings $y_n = x_{2n}$ and $z_n = x_{2n+1}$ with transforms Y_k and Z_k, respectively. From the results in Section 2, Y_k is also O symmetric and Z_k is QO symmetric. Therefore, both X_k and Y_k are strictly imaginary and, from (2.15), Z_k has the form $Z_k = e^{ik2\pi/N}\tilde{Z}_k$, where $\tilde{Z}_k$ is strictly imaginary. Substituting this form into (3.1) and (3.3), we obtain the splitting equations for an O symmetric sequence,

(3.6) $$X_k = Y_k + \tilde{Z}_k, \qquad k = 1, \ldots, N/4,$$

(3.7) $$X_{N/2-k} = Y_k - \tilde{Z}_k, \qquad k = 1, \ldots, N/4 - 1.$$

Since X_k, Y_k, and $\tilde{Z}_k$ are all strictly imaginary, the splitting equations require only $N/2 - 1$ real additions.

3.5. *Splitting Equations for QE Symmetric Sequences.* Let $x_n = x_{N-n-1}$ be a QE symmetric sequence. Let $y_n = x_{2n}$ and $z_n = x_{2n+1}$ be the sequences of length $N/2$ in the splitting algorithm. Recall from Section 2 that these sequences have the intersequence symmetry,

(3.8) $$y_{N/2-n-1} = x_{N-2n-2} = x_{2n+1} = z_n.$$

Using a development similar to that preceding (2.13), we find that

(3.9) $$Z_k = e^{ik4\pi/N}\overline{Y}_k.$$

Substituting (2.14) and (3.9) into (3.1) and (3.3) we obtain

$$\tilde{X}_k = 2\,\mathrm{Re}\left[e^{-ik\pi/N}Y_k\right], \qquad k = 0,\ldots,N/4, \tag{3.10}$$

$$\tilde{X}_{N/2-k} = -2\,\mathrm{Im}\left[e^{-ik\pi/N}Y_k\right], \qquad k = 0,\ldots,N/4-1. \tag{3.11}$$

These equations constitute the splitting algorithm for a sequence with QE symmetry. They imply that the discrete Fourier transform of a sequence x_n of length N with QE symmetry can be computed from the Fourier transform Y_k of a real sequence y_n of length $N/2$ with $N-2$ real multiplications and $N/2-1$ real additions.

At this point, we see that the symmetries R, E, and QE are "closed" in the sense that a sequence with any one of these symmetries splits into two sequences, each of which also has one of the three symmetries. Therefore, the splitting algorithms given above for each of these symmetries can be combined to develop discrete FFT algorithms for sequences that are either E or QE symmetric.

3.6. *Splitting Equations for QO Symmetric Sequences.* Let $x_n = -x_{N-n-1}$ be a QO symmetric sequence. Then $y_n = x_{2n}$ and $z_n = x_{2n+1}$ have the intersequence symmetry

$$y_{N/2-n-2} = x_{N-2n-2} = -x_{2n+1} = -z_n. \tag{3.12}$$

Using a development similar to that preceding (2.13), we can determine that

$$Z_k = -e^{ik4\pi/N}\overline{Y}_k. \tag{3.13}$$

Substituting (2.15) and (3.13) into (3.1) and (3.3), we obtain the splitting algorithm for QO symmetric sequences,

$$\tilde{X}_k = 2i\,\mathrm{Im}\left(e^{-ik\pi/N}Y_k\right), \tag{3.14}$$

$$\tilde{X}_{N/2-k} = -2i\,\mathrm{Re}\left(e^{-ik\pi/N}Y_k\right). \tag{3.15}$$

These equations imply that the discrete Fourier transform of a QO symmetric sequence x_n of length N can be computed from the Fourier transform y_k of a real sequence $y_n = x_{2n}$ of length $N/2$ with $N-2$ real multiplications and $N/2-1$ real additions. The splitting equations that have been developed in this section can be used to define symmetric FFTs for any of the five symmetries R, E, O, QE, or QO.

4. The Splitting Equations for General N.

4.1. *The Splitting Equation for the Complex Transform.* In this section, we extend the results that were given in the previous section to the general case in which N has an arbitrary factor p. We begin this section with a derivation of the splitting equation for the full complex FFT. Proceeding in a manner analogous to that following (2.1), we split the sum on the right side of (1.2) into p sums,

$$X_k = \sum_{q=0}^{p-1} e^{-ikq2\pi/N} \sum_{n=0}^{N/p-1} x_{pn+q}e^{-ikn2\pi/(N/p)}. \tag{4.1}$$

If we define the following p sequences, each with N/p elements,

$$x_{n,q} = x_{pn+q}, \qquad q = 0,1,\ldots,p-1, \tag{4.2}$$

and their transforms

$$Y_{k,q} = \sum_{n=0}^{N/p-1} x_{n,q} e^{-ikn2\pi/(N/p)}, \tag{4.3}$$

then (4.1) takes the form

$$X_k = \sum_{q=0}^{p-1} e^{-ikq2\pi/N} Y_{k,q}. \tag{4.4}$$

Next, we define

$$X_{k,l} = X_{lN/p+k} = \sum_{q=0}^{p-1} e^{-i(lN/p+k)q2\pi/N} Y_{lN/p+k,q}. \tag{4.5}$$

It can be shown that $Y_{lN/p+k,q} = Y_{k,q}$ and therefore,

$$X_{k,l} = \sum_{q=0}^{p-1} e^{-ilq2\pi/p} e^{-ikq2\pi/N} Y_{k,q}. \tag{4.6}$$

This is the splitting equation for a complex sequence in which the complex transform X_k of a sequence x_n of length N is expressed in terms of p transforms $Y_{k,q}$ of sequences $x_{n,q}$, each of length N/p.

4.2. *The Splitting Equations for the Real Transform.* If x_n is real, then only half the sequences $X_{k,l}$ and $Y_{k,q}$ need to be computed. In this case, it can be shown that

$$\overline{X}_{N/p-k,l-1} = \sum_{q=0}^{p-1} e^{ilq2\pi/p} e^{-ikq2\pi/N} Y_{k,q}. \tag{4.7}$$

Equations (4.6) and (4.7) constitute the splitting algorithm for a real sequence x_n. Only half of each Fourier transform is computed, with the result that the real FFT can be computed with half the operations required by the complex FFT. With $p = 2$ these equations are the same as (3.1) and (3.3) and thus constitute a generalization of Edson's algorithm for the case in which N has an arbitrary factor p.

4.3. *The Splitting Equations for E Symmetric Sequences.* If $x_n = x_{N-n}$ is E symmetric, then from (4.2),

$$x_{N/p-n-1,q} = x_{N-pn-p+q} = x_{pn+p-q} = x_{n,p-q}. \tag{4.8}$$

It can be shown that $x_{n,0}$ is E symmetric and therefore, $Y_{k,0}$ is real. If p is divisible by 2, then $x_{n,p/2}$ is QE symmetric and therefore $Y_{k,p/2}$ has the form

$$Y_{k,p/2} = e^{ikp\pi/N} \tilde{Y}_{k,p/2}, \tag{4.9}$$

where $\tilde{Y}_{k,p/2}$ is real. For the remaining q, $x_{n,q}$ has real symmetry and the intersequence symmetry (4.8) implies that

$$Y_{k,p-q} = e^{ikp2\pi/N} \overline{Y}_{k,q}. \tag{4.10}$$

Substituting these results into (4.6) and (4.7), we obtain

$$X_{k,l} = Y_{k,0} + 2 \sum_{q=1}^{p/2-1} \operatorname{Re}\left[e^{-ilq2\pi/p} e^{-ikq2\pi/N} Y_{k,q}\right] + (-1)^l \tilde{Y}_{k,p/2}, \tag{4.11}$$

$$X_{N/p-k,l-1} = Y_{k,0} + 2 \sum_{q=1}^{p/2-1} \operatorname{Re}\left[e^{ilq2\pi/p} e^{-ikq2\pi/N} Y_{k,q}\right] + (-1)^l \tilde{Y}_{k,p/2}. \tag{4.12}$$

Equations (4.11) and (4.12) provide the splitting equations for an E symmetric sequence. If p is odd, a somewhat simpler form is obtained.

4.4. *The Splitting Equations for O Symmetric Sequences.* If $x_n = -x_{N-n}$ is O symmetric, then from (4.2),

$$x_{N/p-n-1,q} = x_{N-pn-p+q} = -x_{pn+p-q} = -x_{n,p-q}. \tag{4.13}$$

It can be shown that $x_{n,0}$ is O symmetric, and therefore $Y_{k,0}$ is strictly imaginary. If p is divisible by 2, then $x_{n,p/2}$ is QO symmetric, and therefore $Y_{k,p/2}$ has the form

$$Y_{k,p/2} = e^{ikp\pi/N}\tilde{Y}_{k,p/2}, \tag{4.14}$$

where $\tilde{Y}_{k,p/2}$ is strictly imaginary. For the remaining q, $x_{n,q}$ has real symmetry and the intersequence symmetries (4.13) imply that

$$Y_{k,p-q} = -e^{ikp2\pi/N}\overline{Y}_{k,q}. \tag{4.15}$$

Substituting these results into (4.6) and (4.7), we obtain

$$X_{k,l} = Y_{k,0} + 2i\sum_{q=1}^{p/2-1} \operatorname{Im}\left[e^{-ilq2\pi/p}e^{-ikq2\pi/N}Y_{k,q}\right] + (-1)^l\tilde{Y}_{k,p/2}, \tag{4.16}$$

$$\overline{X}_{N/p-k,l-1} = Y_{k,0} + 2i\sum_{q=1}^{p/2-1} \operatorname{Im}\left[e^{ilq2\pi/p}e^{-ikq2\pi/N}Y_{k,q}\right] + (-1)^l\tilde{Y}_{k,p/2}. \tag{4.17}$$

Equations (4.16) and (4.17) provide the splitting equations for an O symmetric sequence. If p is odd, a somewhat simpler form is obtained.

4.5. *The Splitting Equations for QE Symmetric Sequences.* If $x_n = x_{N-n-1}$ is QE symmetric, then from (2.14),

$$X_{k,l} = X_{lN/p+k} = e^{il\pi/p}e^{ik\pi/N}\tilde{X}_{k,l}, \tag{4.18}$$

and

$$\overline{X}_{N/p-k,l} = e^{-i(l+1)\pi/p}e^{ik\pi/N}\tilde{X}_{N/p-k,l}, \tag{4.19}$$

where $\tilde{X}_{k,l} = \tilde{X}_{lN/p+k}$ is real. From (4.2),

$$x_{N/p-n-1,q} = x_{N-pn-p+q} = x_{pn+p-q-1} = x_{n,p-q-1}. \tag{4.20}$$

If p is odd, $x_{n,(p-1)/2}$ is QE symmetric and hence $Y_{k,(p-1)/2}$ has the form

$$Y_{k,(p-1)/2} = e^{ikp\pi/N}\tilde{Y}_{k,(p-1)/2}, \tag{4.21}$$

where $\tilde{Y}_{k,(p-1)/2}$ is real. For the remaining q, $x_{n,q}$ has real symmetry and the intersequence symmetries (4.20) imply that

$$Y_{k,p-q-1} = e^{ikp2\pi/N}\overline{Y}_{k,q}. \tag{4.22}$$

Substituting (4.18), (4.19), (4.21), and (4.22) into (4.6) and (4.7), we obtain

$$\tilde{X}_{k,l} = 2\sum_{q=0}^{(p-3)/2} \operatorname{Re}\left[e^{-il(2q+1)\pi/p}e^{-ik(2q+1)\pi/N}Y_{k,q}\right] + (-1)^l\tilde{Y}_{k,(p-1)/2}, \tag{4.23}$$

$$\tilde{X}_{N/p-k,l-1} = 2\sum_{q=0}^{(p-3)/2} \operatorname{Re}\left[e^{il(2q+1)\pi/p}e^{-ik(2q+1)\pi/N}Y_{k,q}\right] + (-1)^l\tilde{Y}_{k,(p-1)/2}. \tag{4.24}$$

Equations (4.23) and (4.24) provide the splitting equations for a QE symmetric sequence.

4.6. *The Splitting Equations for QO Symmetric Sequences.* If $x_n = -x_{N-n-1}$ is QO symmetric, then from (2.15),

$$X_{k,l} = X_{lN/p+k} = e^{il\pi/p}e^{ik\pi/N}\tilde{X}_{k,l} \tag{4.25}$$

and

$$\overline{X}_{N/p-k,l} = -e^{-i(l+1)\pi/p}e^{ik\pi/N}\tilde{X}_{N/p-k,l}, \tag{4.26}$$

where $\tilde{X}_{k,l} = \tilde{X}_{lN/p+k}$ is strictly imaginary. From (4.2),

$$x_{N/p-n-1,q} = x_{N-pn-p+q} = -x_{pn+p-q-1} = -x_{n,p-q-1}. \tag{4.27}$$

If p is odd, $x_{n,(p-1)/2}$ is QO symmetric and hence $Y_{k,(p-1)/2}$ has the form

$$Y_{k,(p-1)/2} = e^{ikp\pi/N}\tilde{Y}_{k,(p-1)/2}, \tag{4.28}$$

where $\tilde{Y}_{k,p/2}$ is strictly imaginary. For the remaining q, $x_{n,q}$ has real symmetry, and the intersequence symmetries (4.27) imply that

$$Y_{k,p-q-1} = -e^{ikp2\pi/N}\overline{Y}_{k,q}. \tag{4.29}$$

Substituting (4.25), (4.26), (4.28), and (4.29) into (4.6) and (4.7), we obtain

$$\tilde{X}_{k,l} = 2i\sum_{q=0}^{(p-3)/2} \operatorname{Im}\left[e^{-il(2q+1)\pi/p}e^{-ik(2q+1)\pi/N}Y_{k,q}\right] + (-1)^l\tilde{Y}_{k,(p-1)/2}, \tag{4.30}$$

$$\begin{aligned}\tilde{X}_{N/p-k,l-1} = -2i\sum_{q=0}^{(p-3)/2} \operatorname{Im}\left[e^{il(2q+1)\pi/p}e^{-ik(2q+1)\pi/N}Y_{k,q}\right]\\ -(-1)^l\tilde{Y}_{k,(p-1)/2}.\end{aligned} \tag{4.31}$$

Equations (4.30) and (4.31) provide the splitting equations for a QO symmetric sequence.

5. Existing Symmetric FFTs.

5.1. *Preliminaries.* In this section, we will review existing algorithms for computing the symmetric transforms. The symmetric sequences are preprocessed into either real or complex periodic sequences that can be transformed using FFTs with software that is generally available. The transform is then postprocessed in order to determine the transform of the symmetric sequence. This approach has the advantage of being relatively easy to implement, but the resulting transforms lack the generality and efficiency of the algorithms presented earlier. The real forms of the symmetric transforms are also presented. They are by far the forms that are most commonly used in practice and provide the basis on which most pre- and postprocessing algorithms are developed.

5.2. *Existing Transforms for R Symmetric Sequences.* Two algorithms for real sequences are described in this subsection as well as in a number of other places including Brigham [2]. Both algorithms are about twice as efficient as the complex transform but less efficient than Edson's FFT which was presented earlier.

We show first that two real sequences y_n and z_n can be transformed in the time that is required by a single complex transform. Define the complex sequence $x_n = y_n + iz_n$ and compute its complex transform X_k. Since the Fourier transform is linear,

$$X_k = Y_k + iZ_k, \tag{5.1}$$

where Y_k and Z_k are the Fourier transforms of y_n and z_n, respectively. Since Y_k and Z_k are complex sequences, they do not correspond to the real and imaginary parts of X_k. Nevertheless, they can be computed from X_k in the following manner. From (5.1),

$$X_{N-k} = Y_{N-k} + iZ_{N-k}. \tag{5.2}$$

Since y_n and z_n are real, Y_k and Z_k are conjugate symmetric, and therefore

$$\overline{X}_{N-k} = Y_k - iZ_k. \tag{5.3}$$

Then from (5.1) and (5.3),

$$Y_k = \frac{1}{2}\left(X_k + \overline{X}_{N-k}\right), \tag{5.4}$$

$$Z_k = \frac{1}{2i}\left(X_k - \overline{X}_{N-k}\right). \tag{5.5}$$

Therefore, two real transforms of length N can be computed from a single complex transform of length N. First, form $x_n = y_n + iz_n$, next compute X_k, and finally compute Y_k and Z_k from (5.4) and (5.5).

The second algorithm for real sequences consists of transforming a single real sequence of length N by pre- and postprocessing around a complex transform of length $N/2$. Let x_n be a real sequence of length N, which is an even integer. Define $y_n = x_{2n}$ and $z_n = x_{2n+1}$ and $w_n = y_n + iz_n$. Next, compute the Fourier transform W_k of w_n. From (5.4) and (5.5),

$$Y_k = \frac{1}{2}\left(W_k + \overline{W}_{N/2-k}\right), \tag{5.6}$$

$$Z_k = \frac{1}{2i}\left(W_k - \overline{W}_{N/2-k}\right), \tag{5.7}$$

where Y_k and Z_k are the discrete Fourier transforms of the sequences $y_n = x_{2n}$ and $z_n = x_{2n+1}$. But X_k can now be computed from the splitting equations (3.1) and (3.3) for a real sequence.

In summary, given a real sequence x_n, we first form $w_n = x_{2n} + ix_{2n+1}$ and compute its discrete complex transform W_k. We next compute Y_k and Z_k from (5.6) and (5.7), and finally X_k from (3.1) and (3.3).

We close this part with the development of the trigonometric representation of the transform of a real sequence. This form will be used in the development of the remaining symmetric transforms, which are also presented in trigonometric form.

If we assume for the moment that N is an even integer, then (1.1) can be written

$$x_n = \sum_{k=0}^{N/2-1} X_k e^{ikn2\pi/N} + \sum_{k=1}^{N/2} X_{N-k} e^{i(N-k)n2\pi/N}. \tag{5.8}$$

Since X_k is conjugate symmetric,

$$x_n = X_0 + 2\sum_{k=1}^{N/2-1} \operatorname{Re}\left[X_k e^{ikn2\pi/N}\right] + (-1)^n X_{N/2}, \tag{5.9}$$

or

$$x_n = \frac{1}{2}a_0 + \sum_{k=1}^{N/2-1}\left(a_k \cos kn\frac{2\pi}{N} + b_k \sin kn\frac{2\pi}{N}\right) + \frac{1}{2}a_{N/2}(-1)^n, \tag{5.10}$$

where

$$a_k = 2\,\mathrm{Re}(X_k) \quad \text{and} \quad b_k = -2\,\mathrm{Im}(X_k). \tag{5.11}$$

Using the fact that x_n is real, it can be shown from (1.2) and (5.11) that

$$a_k = \frac{2}{N}\sum_{n=0}^{N-1} x_n \cos kn\frac{2\pi}{N}, \tag{5.12}$$

$$b_k = \frac{2}{N}\sum_{n=1}^{N-1} x_n \sin kn\frac{2\pi}{N}. \tag{5.13}$$

Equations (5.10), (5.12), and (5.13) comprise the real form of (1.1) and (1.2) for a real sequence x_n. The relation between the complex and the real forms is established by (5.11).

If N is an odd integer, then (5.10) takes the form

$$x_n = \frac{1}{2}a_0 + \sum_{k=1}^{(N-1)/2}\left(a_k \cos kn\frac{2\pi}{N} + b_k \sin kn\frac{2\pi}{N}\right). \tag{5.14}$$

5.3. *Existing Transforms for E Symmetric Sequences.* For $n = 0,\ldots,N-1$, let $x_n = x_{N-n}$ be a real even sequence. From Table 3, we note that X_k is real and hence, from (5.11), $b_k = 0$. If N is an even integer, then from (5.10), x_n has the trigonometric representation

$$x_n = \frac{1}{2}a_0 + \sum_{k=1}^{N/2-1} a_k \cos kn\frac{2\pi}{N} + \frac{1}{2}(-1)^n a_{N/2}. \tag{5.15}$$

Substituting $x_n = x_{N-n}$ into (5.12), we obtain

$$a_k = \frac{2}{N}x_0 + \frac{4}{N}\sum_{n=1}^{N/2-1} x_n \cos kn\frac{2\pi}{N} + \frac{2}{N}(-1)^k x_{N/2}. \tag{5.16}$$

If N is an odd integer, then x_n has the trigonometric representation

$$x_n = \frac{1}{2}a_0 + \sum_{k=1}^{(N-1)/2} a_k \cos kn\frac{2\pi}{N}, \tag{5.17}$$

where

$$a_k = \frac{2}{N}x_0 + \frac{4}{N}\sum_{k=1}^{(N-1)/2} x_n \cos kn\frac{2\pi}{N}. \tag{5.18}$$

We observe that the Fourier transform of an even sequence is the inverse of itself, except for a multiplicative constant. That is, except for a factor of $4/N$, Eq. (5.16) is the same as (5.15), and (5.18) is the same as (5.17).

Assume that an E symmetric sequence $x_n = x_{N-n}$ is given, and we wish to compute its real Fourier transform a_k given by (5.16). The first step is to compute a new sequence e_n given by

$$e_n = (x_n + x_{N/2-n}) - \sin n\frac{2\pi}{N}(x_n - x_{N/2-n}). \tag{5.19}$$

If $N/2$ is also even, then it can be shown by substituting (5.15) into (5.19) that

$$\begin{aligned} e_n = a_0 + \sum_{k=1}^{N/4-1}\left[a_{2k}\cos kn\frac{4\pi}{N} + (a_{2k+1} - a_{2k-1})\sin kn\frac{4\pi}{N}\right] \\ + (-1)^n a_{N/2}. \end{aligned} \tag{5.20}$$

By comparing this equation with (5.10), it can be seen that e_n is an R symmetric sequence of length $N/2$. Hence, we can use the real periodic transform to compute coefficients c_k and d_k such that

$$e_n = c_0 + \sum_{k=1}^{N/4-1} \left(c_k \cos kn\frac{4\pi}{N} + d_k \sin kn\frac{4\pi}{N} \right) + (-1)^n c_{N/4}. \tag{5.21}$$

Equating coefficients in (5.20) and (5.21),

$$a_{2k} = c_k, \qquad k = 0, \dots, N/4, \tag{5.22}$$

$$a_{2k+1} - a_{2k-1} = d_k, \qquad k = 1, \dots, N/4 - 1. \tag{5.23}$$

We can now summarize the algorithm for the transform of an E symmetric sequence:

1. Given the E symmetric sequence x_n, compute e_n from (5.19). This calculation can be made more efficient by computing $e_{N/2-n}$ and e_n at the same time.
2. Use the real periodic transform to compute c_k and d_k of the R symmetric sequence e_n of length $N/2$.
3. Compute a_{2k} from (5.22), and a_{2k+1} by using (5.23) in recurrence form

$$a_{2k+1} = a_{2k-1} + d_k, \tag{5.24}$$

which is initialized by computing a_1 from (5.16).

This algorithm is due to Dollimore [4]. Unlike the algorithms that are presented in the previous sections, N must be even. The requirement that $N/2$ also be even can be eliminated by developing an alternate algorithm that is based on Eqs. (5.14), (5.17), and (5.18). From Eq. (5.15) through (5.18), it can be observed that the transform of an E symmetric sequence is essentially the inverse of itself. Therefore, an alternate algorithm for E symmetric sequences can be developed by reversing the order of the computations given above. The resulting algorithm is due to Cooley, Lewis, and Welsh [3]. This algorithm may not be as attractive since it requires a division by $\sin(2\pi/N)$.

5.4. *Existing Transforms of O Symmetric Sequences.* For $n = 0, \dots, N-1$, let $x_n = -x_{N-n}$ be a real odd sequence. From Table 3 we observe that X_k is strictly imaginary and hence, from (5.11), that $a_k = 0$. If N is an even integer, then from (5.10), x_n has the trigonometric representation

$$x_n = \sum_{k=1}^{N/2-1} b_k \sin kn\frac{2\pi}{N}. \tag{5.25}$$

Substituting $x_n = -x_{N-n}$ into (5.13), we obtain

$$b_k = \frac{4}{N} \sum_{n=1}^{N/2-1} x_n \sin kn\frac{2\pi}{N}. \tag{5.26}$$

If N is an odd integer, then x_n has the trigonometric representation

$$x_n = \sum_{k=1}^{(N-1)/2} b_k \sin kn\frac{2\pi}{N}, \tag{5.27}$$

where

$$b_k = \frac{4}{N} \sum_{n=1}^{(N-1)/2} x_n \sin kn\frac{2\pi}{N}. \tag{5.28}$$

We observe that the Fourier transform of an odd sequence is the inverse of itself, except for a multiplicative factor of $4/N$. That is, except for this factor, Eqs. (5.25) and (5.26) are the same, as are Eqs. (5.27) and (5.28).

Assume that an O symmetric sequence $x_n = -x_{N-n}$ is given, and that we want to compute its Fourier transform b_k given by (5.26). The first step is to compute a new sequence e_n given by

$$e_n = (x_n - x_{N/2-n}) + \sin n\frac{2\pi}{N}(x_n + x_{N/2-n}). \tag{5.29}$$

If $N/2$ is also even, it can be shown by substituting (5.25) into (5.29) that

$$\begin{aligned} e_n = b_1 + \sum_{k=1}^{N/4-1}\left[(b_{2k+1} - b_{2k-1})\cos kn\frac{4\pi}{N} + b_{2k}\sin kn\frac{4\pi}{N}\right] \\ -(-1)^n b_{N/2-1}. \end{aligned} \tag{5.30}$$

By comparing this equation with (5.10), we can see that e_n is an R symmetric sequence of length $N/2$. Hence, we can use the real periodic transform (5.12) and (5.13) to compute c_k and d_k such that

$$e_n = c_0 + \sum_{k=1}^{N/4-1}\left(c_k\cos kn\frac{4\pi}{N} + d_k\sin kn\frac{4\pi}{N}\right) + (-1)^n c_{N/4}. \tag{5.31}$$

Equating coefficients in (5.30) and (5.31),

$$b_1 = c_0, \tag{5.32}$$

$$b_{2k+1} - b_{2k-1} = c_k, \tag{5.33}$$

$$b_{2k} = d_k, \tag{5.34}$$

$$b_{N/2-1} = -c_{N/4}. \tag{5.35}$$

We can now summarize the algorithm for the transform of O symmetric sequences:

1. Given the O symmetric sequence $x_n = -x_{N-n}$, compute e_n from (5.29). This calculation can be made more efficient by computing $e_{N/2-n}$ and e_n at the same time.

2. Use the real periodic transform to compute the Fourier transform c_k and d_k of the R symmetric sequence e_n of length $N/2$.

3. Compute b_{2k} from (5.34), and b_{2k+1} for odd subscripts by using (5.33) in recurrence form

$$b_{2k+1} = b_{2k-1} + c_k, \tag{5.36}$$

starting with b_1 computed from (5.32).

This algorithm is due to Dollimore [4]. Unlike the algorithms that are presented in the earlier sections, N must be even. The requirement that $N/2$ must also be even can be eliminated by developing an alternate algorithm that is based on Eqs. (5.14), (5.27), and (5.28). An alternate algorithm for O symmetric sequences can be developed by reversing the order of the calculations. The resulting algorithm is due to Cooley, Lewis, and Welsh [3]. However, like the transform for even sequences, this algorithm may not be as attractive since it requires a division by $\sin(2\pi/N)$.

5.5. *Existing Transforms for QE Symmetric Sequences.* For $n = 0, \ldots, N-1$, let $x_n = x_{N-n-1}$ be a QE symmetric sequence. We start the development of the real form by substituting (2.14) into (5.9). Assume for the moment that N is even, then

$$x_n = \tilde{X}_0 + 2 \sum_{k=1}^{N/2-1} \operatorname{Re}\left[\tilde{X}_k e^{ik(2n+1)\pi/N}\right], \tag{5.37}$$

or

$$x_n = \frac{1}{2}a_0 + \sum_{k=1}^{N/2-1} a_k \cos k(2n+1)\frac{\pi}{N}, \tag{5.38}$$

where

$$a_k = 2\tilde{X}_k = 2e^{-ik\pi/N}X_k. \tag{5.39}$$

To obtain the forward transform, we begin by splitting the sum on the right side of (1.2),

$$X_k = \frac{1}{N}\sum_{n=0}^{N/2-1} x_n e^{-ikn2\pi/N} + \frac{1}{N}\sum_{n=0}^{N/2-1} x_{N-n-1}e^{-ik(N-n-1)2\pi/N}. \tag{5.40}$$

Substituting (5.39) into the left side of (5.40) and $x_{N-n-1} = x_n$ into the right side, we obtain

$$a_k = \frac{4}{N}\sum_{n=0}^{N/2-1} x_n \cos k(2n+1)\frac{\pi}{N}. \tag{5.41}$$

Equations (5.41) and (5.38) comprise the real form of the discrete Fourier transform and its inverse for a QE symmetric sequence. Their relation with the complex forms (1.1) and (1.2) is given in Eq. (5.39).

For N an odd integer, the real forms of the transforms are

$$a_k = \frac{4}{N}\sum_{n=0}^{(N-3)/2} x_n \cos k(2n+1)\frac{\pi}{N} + \frac{2}{N}(-1)^k x_{(N-1)/2}, \tag{5.42}$$

$$x_n = \frac{1}{2}a_0 + \sum_{k=1}^{(N-1)/2} a_k \cos k(2n+1)\frac{\pi}{N}. \tag{5.43}$$

Assume that the coefficients a_k are given, and we wish to compute x_n from (5.38). The first step is to compute e_k given by

$$e_k = \cos k\frac{\pi}{N}(a_k + a_{N/2-k}) - \sin k\frac{\pi}{N}(a_k - a_{N/2-k}). \tag{5.44}$$

If $N/2$ is also even, then it can be shown by substituting (5.41) into the right side of (5.44) that

$$\begin{aligned} e_k = \frac{4}{N}x_0 + \frac{4}{N}\sum_{n=1}^{N/4-1}\left[(x_{2n} + x_{2n-1})\cos kn\frac{4\pi}{N} + (x_{2n} - x_{2n-1})\sin kn\frac{4\pi}{N}\right] \\ + \frac{4}{N}(-1)^k x_{N/2-1}. \end{aligned} \tag{5.45}$$

By comparing this equation with (5.10), we can see that e_k is an R symmetric sequence of length $N/2$. Hence, we can use the real periodic transform (5.12) and (5.13) to compute c_n and d_n such that

$$e_k = c_0 + \sum_{n=1}^{N/4-1} \left(c_n \cos kn\frac{4\pi}{N} + d_n \sin kn\frac{4\pi}{N} \right) + (-1)^k c_{N/4}. \tag{5.46}$$

Equating coefficients in (5.45) and (5.46),

$$x_0 = \frac{N}{4} c_0, \tag{5.47}$$

$$x_{2n} + x_{2n-1} = \frac{N}{4} c_n, \tag{5.48}$$

$$x_{2n} - x_{2n-1} = \frac{N}{4} d_n, \tag{5.49}$$

$$x_{N/2-1} = \frac{N}{4} c_{N/4}. \tag{5.50}$$

We can now summarize the algorithm for the inverse transform (5.38) of a QE symmetric sequence:

1. Given the coefficients a_k, compute e_k by using (5.44). This calculation can be made more efficient by computing $e_{N/2-k}$ and e_k at the same time.

2. Use the real periodic transform to compute the Fourier transform c_n and d_n of the R symmetric sequence e_k of length $N/2$.

3. Compute x_0 from (5.47), $x_{N/2-1}$ from (5.50), and the remaining x_n from (5.48) and (5.49) in the form

$$x_{2n} = \frac{N}{8}(c_n + d_n), \qquad n = 1, \dots, N/4 - 1, \tag{5.51}$$

$$x_{2n-1} = \frac{N}{8}(c_n - d_n), \qquad n = 1, \dots, N/4 - 1. \tag{5.52}$$

This algorithm is due to Swarztrauber [7]. Unlike the algorithms that are presented in the earlier sections, N must be even. The requirement that $N/2$ be even can be eliminated by developing an alternate algorithm that is based on Eqs. (5.14), (5.42), and (5.43).

The forward transform (5.41) is computed by reversing the steps in the inverse algorithm:

1. Given x_n, compute c_n and d_n from (5.47) through (5.50).

2. Compute the R symmetric sequence e_k from c_n, d_n by using the inverse periodic real transform (5.10).

3. Compute a_k from e_k using the inverse of (5.44), namely

$$a_k = \left[\cos k\frac{\pi}{N}(e_k + e_{N/2-k}) - \sin k\frac{\pi}{N}(e_k - e_{N/2-k}) \right]. \tag{5.53}$$

5.6. *Existing Transforms of QO Symmetric Sequences.* For $n = 0, \dots, N-1$, let $x_n = -x_{N-n-1}$ be a QO symmetric sequence. Assume for the moment that N is even and substitute (2.15) into (5.9),

$$x_n = 2 \sum_{k=1}^{N/2-1} \operatorname{Re}\left(\tilde{X}_k e^{ik(2n+1)\pi/N} \right) + (-1)^n i \tilde{X}_{N/2}, \tag{5.54}$$

or

$$x_n = \sum_{k=1}^{N/2-1} b_k \sin k(2n+1)\frac{\pi}{N} + (-1)^n b_{N/2}, \tag{5.55}$$

where

$$b_k = 2i\tilde{X}_k = 2ie^{-ik\pi/N}X_k. \tag{5.56}$$

To obtain the forward transform, we begin by substituting $x_{N-n-1} = -x_n$ into the second term on the right side of (5.40),

$$X_k = \frac{1}{N}\sum_{n=0}^{N/2-1} x_n e^{-ikn2\pi/N} - \frac{1}{N}\sum_{n=0}^{N/2-1} x_n e^{ik(n+1)2\pi/N}. \tag{5.57}$$

Next we substitute (5.56) into the left side of (5.57) and obtain

$$b_k = \frac{4}{N}\sum_{n=0}^{N/2-1} x_n \sin k(2n+1)\frac{\pi}{N}. \tag{5.58}$$

Equations (5.58) and (5.55) comprise the real form of the discrete Fourier transform of a QO symmetric sequence and its inverse. Their relation with the complex transforms (1.1) and (1.2) is given in (5.56).

For the case in which N is an odd integer, the real forms of the transform are

$$b_k = \frac{4}{N}\sum_{n=0}^{(N-3)/2} x_n \sin k(2n+1)\frac{\pi}{N} \tag{5.59}$$

and

$$x_n = \sum_{k=1}^{(N-1)/2} b_k \sin k(2n+1)\frac{\pi}{N}. \tag{5.60}$$

Given the coefficients b_k, we wish to compute x_n, which is defined in (5.55). The first step is to compute e_k given by

$$e_k = \cos k\frac{\pi}{N}\left(b_k + b_{N/2-k}\right) + \sin k\frac{\pi}{N}\left(b_k - b_{N/2-k}\right). \tag{5.61}$$

If $N/2$ is also even, it can be shown by substituting (5.58) into the right side of (5.61) that

$$\begin{aligned} e_k = \frac{4}{N}x_0 + \frac{4}{N}\sum_{n=1}^{N/4-1}\left[(x_{2n} - x_{2n-1})\cos kn\frac{4\pi}{N} + (x_{2n} + x_{2n+1})\sin kn\frac{4\pi}{N}\right] \\ -\frac{4}{N}(-1)^k x_{N/2-1}. \end{aligned} \tag{5.62}$$

By comparing this equation with (5.10), it can be seen that e_k is an R symmetric sequence of length $N/2$. Hence, we can use the real periodic transform (5.12) and (5.13) to compute c_n and d_n such that

$$e_k = c_0 + \sum_{n=1}^{N/4-1}\left(c_n \cos kn\frac{4\pi}{N} + d_n \sin kn\frac{4\pi}{N}\right) + (-1)^k c_{N/4}. \tag{5.63}$$

Equating coefficients in (5.62) and (5.63), we obtain

$$x_0 = \frac{N}{4}c_0, \tag{5.64}$$

$$x_{2n} - x_{2n-1} = \frac{N}{4} c_n, \tag{5.65}$$

$$x_{2n} + x_{2n-1} = \frac{N}{4} d_n, \tag{5.66}$$

$$x_{N/2-1} = -\frac{N}{4} c_{N/4}. \tag{5.67}$$

We can now summarize the algorithm for the inverse transform (5.55) of a QO symmetric sequence:

1. Given the coefficients b_k, compute e_k by using (5.61). This calculation can be made more efficient by computing $e_{N/2-k}$ and e_k at the same time.

2. Use the real periodic transform (5.12) and (5.13) to compute the Fourier transform c_n and d_n of the R symmetric sequence e_k of length $N/2$.

3. Compute x_0 from (5.64), $x_{N/2-1}$ from (5.67), and the remaining x_n from (5.65) and (5.66), but in the form

$$x_{2n} = \frac{N}{8}(d_n + c_n), \qquad n = 1, \dots, N/4 - 1, \tag{5.68}$$

$$x_{2n-1} = \frac{N}{8}(d_n - c_n), \qquad n = 1, \dots, N/4 - 1. \tag{5.69}$$

This algorithm is due to Swarztrauber [7]. Unlike the algorithms that are presented in the earlier sections, N must be even. The requirement that $N/2$ also be even can be eliminated by developing an alternate algorithm that is based on Eqs. (5.14), (5.59), and (5.60).

The forward transform (5.58) is computed by reversing the steps in the inverse algorithm:

1. Given x_n, compute c_n and d_n from (5.64) through (5.67).

2. Compute the R symmetric sequence e_k from c_n and d_n by using the inverse periodic real transform (5.10).

3. Compute b_k from e_k by using the inverse of (5.61), namely

$$b_k = \left[\cos k\frac{\pi}{N}(e_k - e_{N/2-k}) + \sin k\frac{\pi}{N}(e_k + e_{N/2-k})\right]. \tag{5.70}$$

6. Computational Notes and Summary.

6.1. *Preliminaries.* In this section, we will consider several topics of practical interest. In the first part, we will compare the operation counts of the new with existing symmetric transforms. Next, we consider certain aspects that are related to the implementation of the transforms, including those associated with advanced computer architectures. Finally, we summarize briefly the results presented in this paper.

6.2. *Operation Counts.* We begin this section with Table 4 which contains the operation counts for the various symmetric transforms. This table can be compared with Table 5, which contains the operation counts for the existing transforms that are given in Section 5. The counts given in both tables are valid for $N \geqslant 8$ because different equations are used for splittings in which the length of the sequence is less than 8. These counts do not include multiplications by 1, -1, i, or $-i$.

TABLE 4
Operation counts for the FFT of a sequence of length $N = 2^m$

transform	additions	multiplications
full complex	$3mN - 2N$	$2mN - 4N$
real (Edson's)	$3mN/2 - 5N/2 + 2$	$mN - 3N + 4$
E symmetric	$3mN/4 - 3N/2 + 2m + 1$	$mN/2 - 2N + 2m - 2$
O symmetric	$3mN/4 - 3N/2 + 9$	$mN/2 - 2N + 2m - 2$
QE symmetric	$3mN/4 - 2N + 2$	$mN/2 - N + 2$
QO symmetric	$3mN/4 - 2N + 2$	$mN/2 - N + 2$

A comparison of Table 4 with Table 5 yields the observation that the operation counts for the new algorithms are comparable to those that would be obtained by eliminating the pre- and postprocessing from the existing symmetric FFTs. However, the difference in the operation counts does not explain the total difference in the performance of the algorithms. The new algorithms also require fewer data accesses. For example, the FFT of a sequence of length $N = 4^4 = 256$ would require the data to be accessed four times using a radix 4 FFT. However, the pre- and postprocessing, used by the existing symmetric FFT of an E symmetric sequence, requires three additional data accesses. This 75% increase in the number of data accesses will significantly increase the amount of computing, particularly on a vector or pipeline computer.

TABLE 5
Operation counts for existing symmetric FFTs of length $N = 2^m$

transform	additions	multiplications
real periodic	$3mN/2 - 2$	$mN - 2N - 4$
E symmetric	$3mN/4 + N/2 + 5$	$mN/2 - N + 1$
O symmetric	$3mN/4 - 3N/4 + 1$	$mN/2 - 3N/2 + 2$
QE symmetric	$3mN/4 - N/2 + 4$	$mN/2 - 3N/2 + 2$
QO symmetric	$3mN/4 - N/2 + 4$	$mN/2 - 3N/2 + 2$

The operation counts given in Table 5 were computed as follows. The count for the real periodic transform was obtained from the second algorithm given in Subsection 5.2. It includes the computations in (5.6), (5.7), (3.1), and (3.3), plus the computations required for a complex transform of length $N/2$, computed from Table 4. The counts for the remaining existing transforms include the counts for pre- and postprocessing plus the counts for a real transform computed from Table 4 rather than Table 5. Computing the counts in this manner produces those most favorable to the existing symmetric FFTs.

6.3. *Vectorizing the Symmetric FFTs.* An examination of Table 2 reveals that the programming of the algorithm for the E symmetric, and indeed all symmetric sequences, is relatively straightforward. One need only modify an existing program for real or R symmetric sequences. The first two sequences in a column are computed using the splitting algorithms for E and QE symmetric sequences, respectively. The remaining sequences use the existing splitting equations for R symmetric sequences.

Consider now the implementation of the symmetric FFTs on the recent generation of computers with advanced architectures. The vectorization of the FFTs has been discussed in a number of places, including [5], [8] and [10]. Depending on the method, the vectorization of the symmetric FFT may be unchanged from the complex FFT. The vectorization of multiple transforms can be performed by applying each operation in the FFT to all of the sequences. When this approach is used, the vector lengths are equal to the number of sequences. This method can also be applied to the symmetric FFTs.

The vectorization of a single transform is complicated by the fact that the sequences in Table 1 get shorter as they are split. However, as the sequences get shorter, their number increases. Using the "loop inversion" method described in [8] and [10], we select a vector, either as a sequence or as a collection of the same element from each of the sequences. Therefore, the length of the vector is equal either to the length of the sequences or to the number of sequences, whichever is the greatest. The length of the vectors still decrease initially but then increase after loop inversion.

Once the loop is inverted, only elements from the R symmetric sequences can be included in the vector, since different computations are performed on the remaining symmetric sequences. For example, in Table 2, the number of R symmetric sequences determines the length of the vector after loop inversion, which is equal to $2^{i-2} - 1$ for $i > 2$. Interestingly, this has very little effect on the length of the smallest vector. From Table 6, the minimum length of a vector in the transform of an R symmetric sequence differs by only 1 from that of an E symmetric sequence.

TABLE 6
Minimum vector length in a single FFT of length $N = 2^m$

transform	m even	m odd
complex	$\sqrt{N}$	$\sqrt{N/2}$
R real	$\sqrt{N/4}$	$\sqrt{N/8}$
E, O symmetric	$\sqrt{N/4} - 1$	$\sqrt{N/8}$
QE, QO symmetric	$\sqrt{N/4}$	$\sqrt{N/8}$

Longer vectors can be obtained by using symmetric transforms that are based on FFTs other than the Cooley-Tukey. If the Pease algorithm is used, then all vector lengths are $N/2$ in the computational phase. However, the Pease algorithm requires an additional phase in which the transform is ordered. This phase can be performed in $\log N$ steps using a FORTRAN program in which the vectors again decrease in length to $\sqrt{N/2}$. Alternately, the ordering can be achieved in $\log N$ steps using the compress and merge machine language instructions that are available on pipeline computers.

On the Control Data CYBER-205, where compress operations are significantly more efficient than merges, Fornberg [5] has shown that the Glassman FFT can be implemented using only compress instructions. Two compresses are inserted at each of the $\log N$ steps of the FFT with the result that the transform is ordered and all vectors have length $N/2$. This approach can also be used in the Stockham FFTs, however it is not as attractive on the CRAY-1 where the insertion of the compress (gather) instructions can be more expensive than short vectors.

6.4. *Summary.* In this paper, we have examined existing efficient algorithms for symmetric sequences. The symmetries are those generated by the splittings that occur in the FFT of real even or odd sequences. There are five symmetries generated by the splittings, namely, real, real even, real odd, and the even and odd quarter-wave symmetries. We have used these symmetries to develop new symmetric FFTs that are free of pre- and postprocessing as well as of the restriction that the length of the sequence be an even integer. We have also shown that the amount of computational work is that which would be expected. That is, just as the transform of a real sequence takes half that of a complex sequence, the transform of a real odd or even sequence takes half that of a real sequence.

Although the presentation of the symmetric FFTs has been made in the context of the Cooley-Tukey FFT, the results are applicable to any of the FFTs. That is, symmetric FFTs can also be developed based on the Pease or Stockham autosort algorithms given in [8] and [10]. This is due to the fact that the FFTs differ only in the way in which the intermediate computations are stored. An autosort symmetric FFT can be developed by replacing the splitting equations used in the autosort FFTs with those developed in this paper.

Acknowledgment. The author wishes to acknowledge the helpful comments of the referee, particularly those which led to the discussion at the end of Subsection 6.3.

National Center for Atmospheric Research
Boulder, Colorado 80307

1. G. D. Bergland, "A fast Fourier transform for real-valued series," *Comm. ACM*, v. 11, 1968, pp. 703–710.

2. E. O. Brigham, *The Fast Fourier Transform*, Prentice-Hall, Englewood Cliffs, N. J., 1974.

3. J. W. Cooley, P. A. W. Lewis & P. D. Welsh, "The fast Fourier transform algorithm: Programming considerations in the calculation of sine, cosine and Laplace transforms," *J. Sound Vibration*, v. 12, 1970, pp. 315–337.

4. J. Dollimore, "Some algorithms for use with the fast Fourier transform," *J. Inst. Math. Appl.*, v. 12, 1973, pp. 115–117.

5. B. Fornberg, "A vector implementation of the fast Fourier transform," *Math. Comp.*, v. 36, 1981, pp. 189–191.

6. R. W. Hockney, "A fast direct solution of Poisson's equation using Fourier analysis," *J. Assoc. Comput. Mach.*, v. 12, 1965, pp. 95–113.

7. P. N. Swarztrauber, "The methods of cyclic reduction, Fourier analysis and the FACR algorithm for the discrete solution of Poisson's equation on a rectangle," *SIAM Rev.*, v. 19, 1977, pp. 490–501.

8. P. N. Swarztrauber, "Vectorizing the FFTs," in *Parallel Computations* (G. Rodrigue, ed.), Academic Press, New York, 1982, pp. 490–501.

9. P. N. Swarztrauber, "Software for the spectral analysis of scalar and vector functions on the sphere," in *Large Scale Scientific Computation* (S. V. Parter, ed.), Academic Press, New York, 1984, pp. 271–299.

10. P. N. Swarztrauber, "FFT algorithms for vector computers," *Parallel Computing*, v. 1, 1984, pp. 45–63.

11. P. N. Swarztrauber, "Fast Poisson solvers," *MAA Studies in Numerical Analysis*, Vol. 24 (G. H. Golub, ed.), Mathematical Association of America, 1984, pp. 319–370.

12. C. Temperton, "Fast mixed-radix real Fourier transforms," *J. Comput. Phys.*, v. 52, 1983, pp. 340–350.